Modeling Software Helps Rocket Scientists Go with the Flow

NASA Technology

Rocket science and simplicity don't normally go hand in hand, but that's what NASA had in mind when the Agency developed the Fastrac turbopump in the mid-1990s.

Turbopumps are responsible for sending propellant from the tank to the combustion engine with a high enough pressure to generate the needed thrust, which is accomplished by drawing in low-pressure propellant and compressing it before it's injected into the combustion chamber. The Space Shuttle main engine turbopump was considered one of the most sophisticated reusable engines of its kind. It used a staged-combustion cycle, meaning some of the propellants—liquid oxygen and liquid hydrogen—were burned in preburners to create hot gas to power the turbines and turbopump. The liquid propellants were then injected into the combustion chamber. In addition to requiring four pumps to pressurize the gases, the hydrogen needed to be kept at ultralow temperatures to remain in its mandatory liquid state.

Designed for smaller, less expensive spacecraft, Fastrac was made to run more efficiently and with simpler plumbing and fewer parts. Instead of liquid hydrogen, it uses a highly refined kerosene called Rocket Propellant-1, or RP-1, which is cheaper and easier to handle. The staged-combustion cycle was replaced with the less complicated gas-generator cycle, doing away with the complexities inherent in having to inject exhausted gas from the preburner to the combustion engine.

The goal was to come up with a simpler turbopump, but getting there still required rocket science know-how, namely computational fluid dynamics (CFD), which involves three disciplines: thermodynamics, fluid mechanics, and heat transfer. They're applied to resolve a number of thorny issues, from figuring out how much coolant is needed and ensuring that enough pressure is being generated in the turbine to making sure the RP-1 and liquid oxygen, separated by shafts mere inches apart, do not mix anywhere other than in the combustion chamber, lest an explosion occur.

NASA ushered in an era of simpler, more efficient turbopumps with the invention of Fastrac in the 1990s. To minimize costly test stand launches, the Agency created the Generalized Fluid System Simulation Program, or GFSSP, to analyze computational fluid dynamics. SpaceX used Fastrac as a blueprint for developing its Merlin family of rocket engines for the Falcon 9 launch vehicle, which is responsible for delivering payloads to the ISS.

It's better to iron out most of the kinks by using analytic software rather by than firing up imprecisely designed turbopumps on the test stand, says Marshall Space Flight Center aerospace technologist Alok Majumdar. Aside from the very real potential for major accidents, tests are also expensive. "A single test of the Space Shuttle main engine cost a million dollars, just for a couple minutes of firing," he says. "You still need tests, but if you can use your analytical tool to model the performance of the rocket engine, instead of 10 tests, maybe you can limit it to 2 or 3. It's a big cost savings."

But NASA was at a disadvantage because the Agency did not have access to a general-purpose code for performing such analyses. Aerospace companies created their own proprietary software and, understandably, kept the technology close to their vests. "They would say, 'Give us a contract, and we'll use our tool to give you the numbers,'" Majumdar says. "Our designers wanted a tool so they could make hundreds of runs. They didn't want to depend on somebody from the outside."

Technology Transfer

To meet that need, in 1994, just as Fastrac was getting underway, Majumdar began developing the Generalized Fluid System Simulation Program, or GFSSP, which was first used by the Agency's analysts in October 1996. NASA employees used the software for the Fastrac turbopump, and even though the program was cancelled in the early 2000s, the turbopump became the predecessor for others like it, such as SpaceX's Merlin rocket engines,

An artist's concept showing Virgin Galactic's LauncherOne spacecraft just after stage separation. Engineers are designing LauncherOne to be able to launch small satellites in space and are using GFSSP for tank-sizing and pressurization analysis.

which power the company's Falcon 9 commercial launch vehicle.

In the late 1990s, NASA made GFSSP available for free to other Government agencies and to Government contractors. Meanwhile, turbomachinery design company Concepts NREC, based in White River Junction, Vermont, licensed the technology and now sells the code as part of its own software package.

Benefits

GFSSP started out as a basic program, capable of computing only steady-state fluid models, but numerous functions were, and are still, being added and released in newer versions. For example, Majumdar and his team have developed code for analyzing solids via heat conduction, radiation, and the convective heat transfer between solids and fluids. Evaluating multilayer insulation for the tanks so gases are kept cool enough to remain in liquid form is another application, as is the ability to analyze mixtures that contain different liquid components.

Among its current users is Virgin Galactic, which is in the midst of developing an affordable rocket, called LauncherOne, aimed at the small satellite market. Kim Betker, a propulsion systems engineer with the company based in Mojave, California, says the software has proven useful for tank-sizing and pressurization system analysis, among other stage-level designs. "It's sometimes hard to figure out a spec for a component until you can see how it affects the other parts it's connected to," she says. "GFSSP has been really good about letting us put a bunch of components in a series to see how they all work together. That lets us figure out a good operating envelope for each component."

Even a mining company is making use of GFSSP. "They wanted to remove methane from their mine shafts and needed to figure out the types of compressors needed to get the gas to an acceptable concentration so it didn't pose an explosion risk," Majumdar says. "Basically, it's looking at flow distribution through a network. That you can use a rocket engine code to do mine ventilation is very striking to me."

Aside from its versatility, Majumdar says GFSSP is relatively easy to use compared with other CFD codes that require expertise in various programming languages. "We do not expect that only a highly sophisticated numerical analyst will run this code," he says, adding that any engineer who has some basic knowledge of thermo-fluid dynamics can use it.

Even though GFSSP won NASA's Software of the Year award back in 2001, Majumdar and his team show no signs of letting up on enhancing its capabilities. In addition to improving pre- and post-processor speeds, the program's next iteration, version 7.0, will include a separated phase model, which looks at mixtures of gas-liquids that move at different speeds, instead of assuming them to be homogeneous mixtures where all phases move with the same speed.

The classic case for this sort of problem-solving, Majumdar says, involves hydrogen rocket propellant. As liquid hydrogen travels from the tank to the turbopump, as soon as it makes initial contact with piping it turns to vapor because of the rise in temperature. As the pipe is cooled, the vapor will become a mixture of vapor and liquid before it returns to a full liquid. "Once it becomes a full liquid, you don't have any problems, but when there are two phases with cryogenic propellant, they don't move at the same velocity," he explains. "You need to have a modeling technique that can tease out what effects that will have on the overall system."

Such an analysis will be another big step for software that continues to impress nearly 20 years after its inception.

"It's such a complex process, and you're solving so many equations, which are coupled with each other and are highly nonlinear," Majumdar says. "But over the years we have done a lot of work on our solvers, and we now have the confidence that our solvers can handle these complexities." ❖